

California Energy Commission Award No. 500-10-052
National Lab Buildings Energy Efficiency Research Projects
LBNL EF87EE

**Task 2.11: Improved Audio-Video Efficiency
Through Inter-Device Control**

**Deliverable 3: Use Cases, Analysis Method, and
Candidate Device Behavior Model Report**

Bruce Nordman, Iris Cheung

December 21, 2012

TABLE OF CONTENTS

Background	4
Introduction	4
Design Principles	4
Concepts	5
Device	5
Power State	5
Source	6
Sink	6
Intermediate	6
Stream	6
Sleeping stream	7
Action	8
Transition states	8
Use Cases	9
One-Device Use Cases	11
Two-Device Use Cases	13
Failure Cases	17
Three-Device Use Cases	18
Stream-focused Use Cases	19
Use Case Summary	20
Behaviors	20
Other Considerations	22
Multiple streams	23
Named streams	23
Multiple sinks and/or sources	23
Creating streams	24
Changing stream structure	24

Failure	24
Occupancy sensors	25
Emergency broadcasts	25
Diverging audio and video	26
Sleeping intermediate devices	26
HDMI switches	26
Legacy Devices	26
Summary.....	27
Acknowledgement.....	28
References	28

Background

This document is the third deliverable for Task 2.11, Improved Audio-Video Efficiency Through Inter-Device Control. The research conducted in this task comprises one element of the National Lab Buildings Energy Efficiency Research Projects, CEC Award No. 500-10-052. The objective of this task is to save substantial amounts of energy by creating a technology standard for how inter-connected audio/video (A/V) devices manage their own power state. The goal is that the technology standard will be incorporated into future products and communication standards.

Introduction

The purpose of this report is to outline use cases of A/V devices, an analysis method, and a candidate device behavior model. This document begins with design principles that have guided the development of the content of this project. The use cases presented are derived from the concept of audio/video (A/V) content streams that have a persistent existence in a local network, being “asleep” when not active. Elements of the use cases are then summarized in device behavior models. Finally, considerations with potential complications are explored. The discussion is abstracted from any specific link or network technology, and does not address how these actions could occur, only what happens. The intent is to layer these concepts on top of existing and future technologies. This discussion is concerned principally with content streams passed over HDMI (High Definition Multimedia Interface) and/or over IP (Internet Protocol) network links.

This analysis builds on the device capabilities found in the first result of this project (Nordman, 2012A), and the technology capabilities described in the second (Nordman, 2012B). It then sets the stage for considering what capabilities need to be added to existing technologies, and what new standards, if any, are needed.

Design Principles

The idea of sleeping streams is a proposed solution to the problem of A/V device coordination, particularly regarding control of power state. Sleeping streams are based on several guiding principles for the architecture of the solution, specifically:

- **No central control.** No single device is required to coordinate the function of other devices. Consequently, there is no possible single point of failure.
- **Inform, not command.** Devices inform each other about their power and functional state (or rather, indirectly by the state of the stream). They do not send each other commands to go to a particular power or functional state. From the desired functional outcomes communicated, devices can determine what power and functional state they need to be in.

- **Streams at core.** Named content streams (and their characteristics) define the outcomes that each device helps to implement.
- **Build on success.** Technologies with wide market penetration today have a better chance than peripheral technologies of continuing to be used in the future. It is easier to add to existing technologies than to create an alternative structure. Web browsing is a good example of this.
- **Use the sleep metaphor.** In many other contexts, the idea of a device or link or other electronic entity being asleep has provided clear terminology for both technical audiences and ordinary users of products.

These principles are derived from experience with technology development and deployment and are consistent with the “Guiding Principles” for energy efficient network technology and policy first outlined in 2007 by the IEA (Nordman et al., 2007).

Concepts

The use cases in this document are based on concepts described below, most of which are commonly used in discussions of A/V devices. The concepts of streams being asleep and the design principles are new to this presentation, and some of the details and assumptions about use cases may be novel as well.

Device

A device is a unified entity which communicates with other devices and can perform one or more A/V functions — principally acquiring, processing, storing, or displaying A/V content. A device is either a source, a sink, an intermediate, or some combination of these.

IP network infrastructure devices (e.g. switches, routers) are not considered part of content streams for several reasons. They are always on, so their power state is not affected by the presence or absence of a stream. They do not interact with the content of the stream (they may be cognizant of “quality of service” requirements of a stream but that is accomplished by lower layer protocols and invisible to this discussion). Finally, the path through a network is not necessarily fixed.

Power State

A device can be On, in a Sleep mode, or Off. A stream can also be in one of these same three basic states. A device that is off can only exit that mode through a power command; a stream that is Off does not exist. A device that is asleep may wake for a large number of reasons, mostly related to functionality. A stream that is asleep may be woken by any of the devices involved, or select other devices.

Using the term “power state” for a non-physical entity like a stream may seem peculiar, but it is useful to parallel the concept for devices. Also, a better term is yet to be identified.

Source

A source is a device that is an origin of A/V content in a local network. A local device with content delivered from outside the local network is considered a source (this occurs for broadcast TV and for Internet video streaming). Examples of sources include set-top boxes (external content), static media players (e.g. DVD player), local recording devices, and local cameras.

Sink

A sink device is a destination of A/V content in a local network, most commonly a display (audio and/or video), but can also be a recording device, or one that sends the content out of the local network, usually out to the Internet (e.g. a SlingBox¹).

The most common sink is a television. An audio/video receiver (AVR) can be a sink for audio if it sends signals to unpowered speakers. A recording device is a sink when recording.

Intermediate

An intermediate is a device which both receives and sends out content. An intermediate can also include source and/or sink functionality. The most common intermediate is an A/V Receiver, which (mostly) acquires A/V content from one of its inputs and sends it on to a television. A key common function of an AVR is to send out the audio portion to (unpowered) speakers so that it also functions as a sink for audio. An HDMI switch is another example of an intermediate.

An intermediate device sits between a source and sink, and there can be none, one, or many in a stream.

Stream

Wikipedia states that a stream is “a sequence of data elements made available over time” (“[http://en.wikipedia.org/wiki/Stream_\(computing\)](http://en.wikipedia.org/wiki/Stream_(computing))”). An A/V stream is then a sequence of data that represents audio and/or video content that occurs in “real-time” — that is, each minute of content is streamed over a minute of elapsed time. Usually A/V streams are displayed real-time for immediate enjoyment. Some streams go only to recording devices, using the same data transfer mechanisms as user-destined streams. A non-real-time data transfer (e.g. downloading a song on iTunes) is not considered an A/V stream in the context of this report. Some protocols buffer content to increase the quality of the result but retain real-time ultimate delivery.

A stream is also an association among a group of devices, and can be visualized as having connections among the devices, with each link having a direction of content flow. A basic

¹ A SlingBox is a brand name of a device that takes in local A/V content, originally analog video but now also HDMI, and sends it to wherever you are on the Internet. This enables watching local TV from elsewhere, or seeing a video you recorded on your home DVR when you are not at home.

stream has a single source, a single sink, and zero or more intermediate devices. However, multiple sources can be combined into a single stream (devices can have multiple inputs), and streams can also be forwarded to multiple devices simultaneously, e.g. to multiple displays. Audio and video do not need to be routed together (displays and/or speakers may be separate devices), and each link the stream may have attributes such as resolution, volume, etc.

The content in a stream can be continuous (e.g. broadcast TV or game console play), or can be of a fixed, limited size (e.g. DVD playing, disk recording, or door camera² session).

Figure 1 shows the elements of a basic stream, with content flowing from left to right. The dashed lines, for any content flowing into or out of the local network, is not part of the stream (as only the single involved device needs to know that that transfer is occurring). Many streams do not involve content coming in from outside and only a few have it going out. While the figure shows three devices, a stream can involve only two, or even just one.

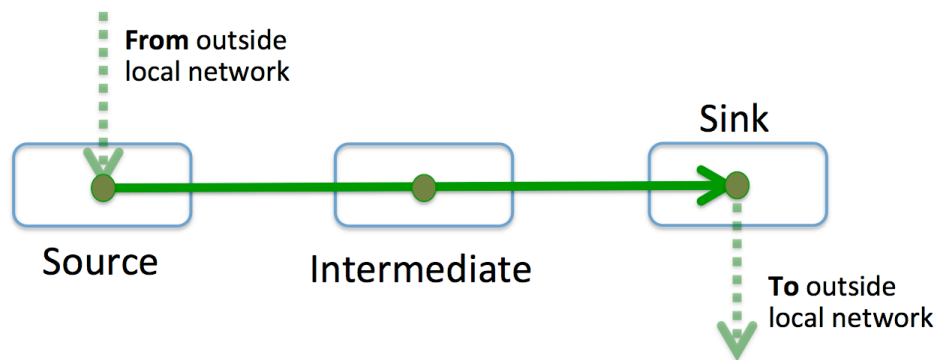


Figure 1. Basic stream structure and key components



Figure 2. Example stream

Sleeping stream

An A/V stream today either exists and is active (pausing is considered to be still part of active), or it is terminated and does not exist at all. The proposed idea of a sleeping stream is one that is not active but whose existence — the devices and settings involved — is retained in the network. Putting aside implementation details, when a stream awakens, then each involved

² A door camera is a networked camera at a front door operated in conjunction with a doorbell to enable someone to see who is visiting. They are particularly useful in apartment buildings where the door can be far from the living quarters.

device will also wake up (unless it is off, or unless it was already on) and the stream will recommence delivering content. When a stream is put to sleep, devices that are part of it often go to sleep (or they may be engaged in unrelated activity and so stay awake).

Action

Events in an A/V system are initiated by one of three types of actions: user-initiated, device internal actions, or other actions. Common user-initiated actions include: powering a device up or down, starting or stopping content, or changing the input source of an A/V device. Device internal actions include timers activating (initiating or terminating a content stream or changing power state), or encountering the end of “fixed” content (see below). An example of an “Other” action is a device disappearing from the local network.

Usually (perhaps always) it does not matter what caused an event, just that it occurred. Users can activate the device directly, with any remote control (the device’s remote or one from a different device), or relayed via a second device.

Transition states

Devices have temporary states that occur while they transition between the three long-term stable power states. Streams similarly will need some transition states. The transitions enable the correct choreography of devices and provide some delay times in cases of uncertainty.

Figure 2 shows a diagram of our proposed states for a stream. On, Sleep, and Off are basic long-term stable states. “Going To Sleep” (GTS), Pause, and Wake are transition states of limited time duration. The stream will know how long it can stay in the state. Devices are likely to have the same transition states.

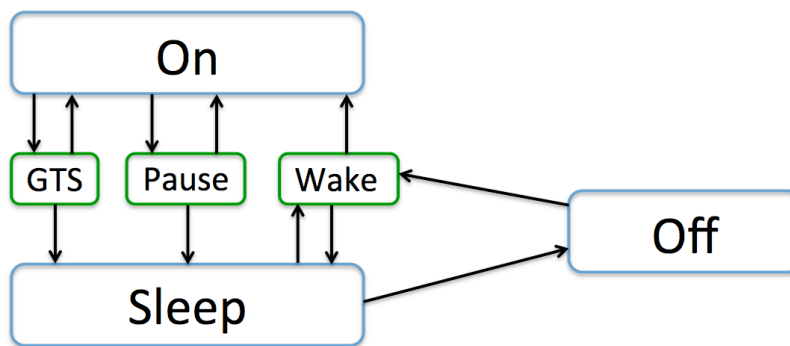


Figure 2. Stream power states

A stream stays in Wake just long enough for all devices involved to become fully On. If a necessary device fails to come on after a specified time duration, the stream wake will fail and it will go back to Sleep. A possible typical duration is 30 seconds.

When a stream is Paused, the expectation is that it will most likely return to full On, and only transition to sleep if it stays in Pause for too long. A possible typical duration for this state is 15 minutes.

When a stream goes into GTS mode, the expectation is that it will soon go to Sleep, but leave the option open for a finite period of time to resume being fully on with minimal delay. A possible typical duration for this state is 30 seconds.

The distinction between the Pause and GTS states is where the stream is heading -- this is determined by the long-term stable state being targeted.

Streams do not actually have an Off state; rather, they drop out of Sleep by being deleted, and in the other direction, are created.

The only way for a device to enter or exit the Off state is with a power command³. A device turning On from the Off state will transition through the Wake state to reactivate or join any desired streams. Powering down to Off in a managed way will usually start at the On state, and will go through GTS, and Sleep, on the way to the Off state, to ensure that streams are put to sleep in an organized, managed way.

While these concepts are part of communication protocols, they also need to be understood by people using A/V systems. These terms need to be considered, and a decision made on whether these are the clearest, most transparent terms to use to describe and define user interfaces.

Use Cases

This discussion presents a high-level summary of use cases for audio/video devices as they create content streams and manage their power state. Aspects of A/V streams that do not affect the power state of devices are not considered here. These use cases are independent of any particular technology to move data.

Typically, use cases are created and then standards are developed to cover the use cases. In this document, the use cases are verbal descriptions which explain how the sleeping stream concept applies, with details for each use case. Thus, use cases are used here somewhat differently than they are in most standards development processes. The use cases also incorporate other important concepts that are needed, such as delay timers, etc.

These use cases are derived from existing products and usages, and will likely be applicable to many new products. Eventually we may need to develop additional use cases or extensions to cover emerging technologies and new functionalities.

The goal of this listing is to consider the desired actions of all types of A/V devices as these behaviors affect power state, and to understand what behaviors the devices must implement. The scenarios that we consider move from one device, to two, and to three. Scenarios with more than three devices have multiple intermediates and do not appear to introduce any new behaviors so it is unnecessary to include any of these. We also do not present cases with three

³ Another way to get to Off is to unplug a device. This is not recommended but devices need to account for this happening as it inevitably will.

or fewer devices that do not introduce new behaviors (behaviors not already covered by the use cases already presented).

Each case involves a set of devices, an initial set of power states, an initiating action (not numbered), a sequence of events that follow, and a final set of power states. Note that “To Sleep” is an action to send a device to the go to sleep state. “Sleep” is a stable power state that a device is in at the beginning or end of the use case.

Below is a sample use case with two devices (DVD player and television) and one stream.

Example: DVD player powers up

Step	DVD	TV	Stream	Comments
START	Sleep	Sleep	Sleep	
				DVD power-up command (manual or internal timer) or manual play command
1	Wake			
2			Wake	DVD wakes up last stream it participated in
3		Wake		Stream involves TV so TV must power up
4		Input		Change Input (If necessary)
5	Play		On	Only after both devices fully wake (only applies to fixed streams)
END	On	On	On	

Note: Case where DVD was Off and is manually powered-up is equivalent.

The first and last rows are the power states of the devices and stream; the unnumbered row is the initiating action; intermediate rows are actions taken by each device. While the steps in the table above are in sequence, the timing is not specified (other scenarios include some steps with specified timings). The mechanisms by which the events described actually happen is not addressed here — just the result.

On waking, a device first checks to see if there are any streams it is part of that are awake or in the process of waking, and then joins these. If none, then most devices will wake the last stream they were participating in before the most recent power-down.

In Step 5 above, the Play action only applies to fixed sources; continuous sources simply move content as soon as they can. Thus, this use case embodies two very similar use cases, with step 5 dropped from one.

In the use cases, a stream often moves from one of the transition states (To-sleep, Pause, or Wake) to the long-term stable state at the end of the sequence with this transition not separately listed. In the example above, the stream becomes On in Step 5, but for a continuous stream there is no step 5 and the transition to On would be implicit in the END state being On.

If the DVD player had been Off instead of in Sleep, and was turned On rather than woken up, the rest of the scenario is exactly the same, so no use case for starting with Off will be included.

Today's DVD players have just one output, to a single sink, but in a network context there may be many sinks available, so identifying the previously active stream as the operation to implement is needed to know which sink to send content to. This is analogous to a TV or AVR switching its input.

These examples only mention one content stream, but devices may be involved in more than one at the same time. These are not common today but will become more so with increasing use of network technology. A computer that displays two different video streams in different windows is involved in multiple streams. When a second stream is involved, then a device may already be fully on when it gets a request to wake up; the request is safely ignored. Similarly, when a device is shown as going to sleep in these use cases, the device might not do so if it is also involved in a second stream.

In the example above, specific devices are listed, but the general sequence applies to any source and to any sink. The cases below are presented as any generic source and any generic sink.

One-Device Use Cases

Use cases that involve a single device are best understood as a TV receiving broadcast television signals. Having the concept of a stream is uninteresting with only one device but is included for completeness and consistency with the other cases. The device acts as both the source and sink of the stream.

1-1) Device Powered Up

Step	Device	Stream	Comments
START	Sleep	Sleep	
			Power up from timer or manually (button or remote)
1	Wake		
2		Wake	
END	On	On	

Note: Case where Device was Off and is manually powered-up is equivalent.

1-2) Stream Woken

Step	Device	Stream	Comments
START	Sleep	Sleep	
			Stream woken
1		Wake	
2	Wake		
END	On	On	

In Case 1-2, some other device knows of the existence of the stream and activates it, which then causes the device to wake. Further stream-oriented scenarios are provided below. Case 1-2 is essentially a subset of S-1 (below) but included here to show the alternate way to initiate stream operation.

1-3) Device Powered Down (timer, manual, or presence of a signal)

Step	Device	Stream	Comments
START	On	On	
			Power down due to timer or manual control
1		To Sleep	
2	To Sleep		
END	Sleep	Sleep	

1-4) Device Powered Down (signal or occupancy)

Step	Device	Stream	Comments
START	On	On	
			Lack of signal or no occupancy
1	Overlay		Display warning (x minutes)
2		To Sleep	
3	To Sleep		
END	Sleep	Sleep	

Cases 1-4 and 1-5 include a visual overlay on the video stream to warn the viewer that a power state change is imminent. A possible duration for the overlay before going to sleep is five minutes.

1-5) Device Put to Sleep (auto-power down)

Step	Device	Stream	Comments
START	On	On	
			Period of inactivity (x hours)
1	Overlay		Display warning (x minutes)
2		To Sleep	Auto-powers down
3	To Sleep		
END	Sleep	Sleep	

Case 1-5 includes a “period of inactivity,” such as no interaction from the remote control to change channel or volume (some devices use a four-hour timer for this). This is an aspect of use cases beyond the simple sequence of events. A timer expiring or manual control can be executed immediately.

Two-Device Use Cases

For two devices, the number of scenarios increases significantly. The general mechanism that devices use to effect other devices is indirect, by affecting the stream, which in turn changes the state of the other devices.

2-1) Source powered up

Step	Source	Sink	Stream	Comments
START	Sleep	Sleep	Sleep	
1				Source manual power-up, manual play, or internal timer
2	Wake			
3			Wake	Source wakes up last stream in which it participated
4		Wake		Stream involves Sink so Sink powers up
5		Input		Change Input (If necessary)
6	Play		On	Only after both devices fully wake (only applies to fixed streams)
END	On	On	On	

Note: Case where Source was Off and is manually powered-up is equivalent.

2-2) Sink powered up

Step	Source	Sink	Stream	Comments
START	Sleep	Sleep	Sleep	
1				Sink manual power-up or internal timer
2		Wake		Sink wakes up last stream in which it participated
3			Wake	
4	Wake			
5	Play			Only after both devices fully wake (only applies to fixed streams)
END	On	On	On	

Note: Case where Sink was Off and is manually powered-up is equivalent.

2-3) Fixed Source ends

Step	Source	Sink	Stream	Comments
START	On	On	On	Source is playing
				Fixed source ends
1	Menu			Source content finishes; display menu for X minutes
2			To Sleep	No input so source puts stream to sleep
3	To Sleep			
4		To Sleep		
END	Sleep	Sleep	Sleep	

Note: Case where Sink ends (recording finished) is equivalent.

2-4) Source paused

Step	Source	Sink	Stream	Comments
START	On	On	On	Source is playing
				Source paused
1	Pause			Wait X minutes
2			To Sleep	
3	To Sleep			
4		To Sleep		
END	Sleep	Sleep	Sleep	

2-5) Sink switched away from source

Step	Source	Sink	Stream	Comments
START	On	On	On	Source is playing
				Sink switched to different source
1			To Sleep	
2	Pause			
3	To Sleep			Source goes to sleep after 15 minutes
END	Sleep	On	Sleep	

2-6) Sink powered down

Step	Source	Sink	Stream	Comments
START	On	On	On	
				Sink receives trigger to power down (manual or timer)
1			To Sleep	Sink tells stream to sleep
2		Sleep		
3	Pause			
4	To Sleep			Source goes to sleep after x minutes
END	Sleep	Sleep	Sleep	

2-7) Source powered down

Step	Source	Sink	Stream	Comments
START	On	On	On	
				Source receives trigger to power down (manual or timer)
1			To Sleep	
2	To Sleep			
3		To Sleep		Sink receives no signal — powers down after a period of time
END	Sleep	Sleep	Sleep	

2-8) Sink switched to Source

Step	Source	Sink	Stream	Comments
START	Sleep	On	Sleep	Source and stream initially asleep
				Sink switches input to Source
1		Change Input		
2			Wake	
3	Wake			
4	Play		On	Only after both devices fully wake (only applies to fixed streams)
END	On	On	On	

2-9) Fixed Sink finishes

Step	Source	Sink	Stream	Comments
START	On	On	On	
				Fixed sink ends
1		To Sleep		Sink goes to sleep
2			To Sleep	Stream goes to sleep
3	To Sleep			
END	Sleep	Sleep	Sleep	

Failure Cases

Some cases do not end with a clear “success” in apparent user intent. These are discussed in more detail below but two examples are shown here for reference.

F-1) Failure on power on

Step	Source	Sink	Stream	Comments
START	Sleep	Off	Sleep	
				Source receives signal for manual power-up, manual play, or internal time
1	Wake			
2			Wake	Source power-up wakes up last stream in which it participated
3		-----		FAIL: Stream involves Sink but Sink is off
4	Error		To Sleep	User gets error message
5	To Sleep			After time of no function Source sleeps (5 min)
END	Sleep	Off	Sleep	

Note: The user may turn on the sink while the error is displayed in which case the stream should activate.

F-2) Failure while active

Step	Source	Intermediate	Sink	Stream	Comments
START	On	On	On	On	
1					Receiver drops off the network
2		Unknown			Can no longer communicate to receiver
3	Pause				Only if it's a fixed source
4				Pause	
5				To Sleep	After waiting a period of time
6	To Sleep		To Sleep		Stream involves source and sink so they power down
END	Sleep	Unknown	Sleep	Sleep	

Three-Device Use Cases

The next set of scenarios is those in which three devices are involved, one of which is an intermediate. The addition of an intermediate to any of the two-device use cases adds nothing except that the intermediate will also wake up or go to sleep when the stream does so. It also may need to change input as the sink does if the stream does not use the current input. Thus, only new cases that three devices raise are presented.

3-1) Intermediate powered up

Step	Source	Intermediate	Sink	Stream	Comments
START	Sleep	Sleep	Sleep	Sleep	
					Receiver manual power-up, manual play, or internal timer
1		Wake			Receiver power-up wakes up last stream in which it participated
2				Wake	
3	Wake		Wake		Stream involves TV and DVD so they power up
4		Input	Input		Change Input (If necessary)
5	Play			On	Only after both devices fully wake (only applies to fixed streams)
END	On		On	On	

3-2) Intermediate powered down

Step	Source	Intermediate	Sink	Stream	Comments
START	On	On	On	On	
					Receiver powers down
1				To Sleep	Receiver power-down makes stream to go to sleep
2		To Sleep			
3	To Sleep		To Sleep		Stream involves source and sink so they power down
END	Sleep	Sleep	Sleep	Sleep	

Stream-focused Use Cases

The final set of scenarios is those in which the stream is addressed directly, rather than indirectly via one of the involved devices. This creates the following two cases. It does not matter how many devices are involved in the stream.

S-1) Stream woken

Step	All Involved Devices	Stream	Comments
START	Sleep	Sleep	Both devices and stream asleep
			Stream woken
1		Wake	
2	Wake		Stream wakes up all involved devices
END	On	On	

S-2) Stream to sleep

Step	All Involved Devices	Stream	Comments
START	On	On	
			Stream sent to sleep
1		To Sleep	
2	To Sleep		Sleeping stream enables all involved devices to go to sleep
END	Sleep	Sleep	

Use Case Summary

As these cases show, it is possible to accomplish inter-device power control solely through communication about stream states. This enables distributed and flexible power control and avoids erroneous inter-device commands.

The sequences above are enabled but not required. For example, some scenarios specify that on waking, a device should resume the last stream in which it was participating before going to sleep. A device could simply power up without engaging a particular stream if it had reason to think that was the desired behavior. This simply subsets the behaviors defined above, rather than expanding on them.

Behaviors

The use cases above describe the collective behavior of a set of entities (devices plus a stream). It is possible to extract the behaviors of the individual entities from these, re-sort them, and then eliminate duplicates. In Tables 1-4 are all the behaviors from the above use cases.

The behaviors are first organized by device type (source, sink, intermediate, and stream), then by the power state each device is in (on, sleep, off), then one or more behaviors that the device implements in this state. In these behaviors, there is first an initiating action, then one or more responses to that action.

For example, in the first case, a source device is fully on and a stream is active. Then, the device is notified that the stream is going to sleep. As the source device, if the content is fixed then it pauses and remembers the place in the stream it was at. If the device is involved in no other streams, then after waiting a short period of time, the source device can itself go to sleep. Note that if a source is paused and not resumed for a defined period of time (the behavior on the right), then that triggers a stream behavior to go to sleep, which then notifies each device, including the source, which then triggers the first behavior. Thus, any sequence of activities is likely to include multiple discrete behaviors. The individual behaviors are then the smallest unit of action.

Table 1. Source Behaviors

On	
Notified stream to go to sleep	Source paused
- Pause stream if fixed	- Wait for X time
- Go to sleep	- Tell stream to sleep
<i>Fixed Streams only</i>	
Fixed content ends	"All devices ready" signal from stream
- Menu for X time	- Play content
- Tell stream to sleep	
Sleep	
Powered up	
- Wake self	
- Wake stream	

Table 2. Sink Behaviors

On	
Notified stream to go to sleep	Switched to different source/stream
- Go to sleep	- Put old stream to sleep
	- Wake new stream
Powered down	Switched to different input
- Tell stream to go to sleep	- Tell old stream to go to sleep
	- Wake new stream
Fixed sink ends	
- Tell stream to go to sleep	
Sleep	
Notified that a stream is waking	Powered up
- Wake self	- Wake self
- Change input (if needed)	- Wake stream

Table 3. Intermediate Behaviors

On	Powered down - Tell stream to go to sleep
Sleep	Powered up - Wake self - Wake stream - Change input if needed

Table 4. Stream Behaviors

On	Notified to go to sleep - Tell all devices it is going to sleep	Unexpected power loss - Stream notices device missing - Display message for X minutes - Put stream to sleep
Sleep	Notified to wake - Tell all device in stream that it is waking - Notice when all devices ready — announce (mainly for fixed)	

Table 5. Failure Behavior

Failure
Some devices off - Put stream to sleep

Other Considerations

The use cases and behaviors above cover ordinary usage, but there are others that will occur that also need to be addressed. These are listed below in no particular order.

Multiple streams

There may be more than one stream active in an A/V network. This is not common today but can be expected in the future. In this case, when a stream initiates waking a device, it may already be awake and active and so not need to wake up. If the device can implement both streams simultaneously, it may, or will need to choose which stream to implement and cause one stream to fail. Conversely, when a stream goes to sleep, if an involved device is involved in other streams, it will not go to sleep.

Named streams

Since today's A/V systems generally have only one content stream existing at a time, there is no need to name streams. However, with multiple streams and persistent streams, then both devices and people need to have a way to uniquely identify them. For people, presumably this would be some sort of text name. This name would be established when the stream is created, a process we do not need to consider. However, the name might be useful to include in warning or error messages, e.g. when a stream is going to go to sleep due to inactivity, absence, or failure.

Multiple sinks and/or sources

While content streams today have usually just a single source and a single sink, in the future more complex streams are likely to burgeon. Relaying content to multiple displays is particularly a likely near-term use, for example so that people at home can visit the kitchen without missing part of a sports event. Whether the displays are in series or in parallel changes the semantics of their operation. For example, if a display in a kitchen gets a stream from the source, then changing the source for a TV in a different room would not affect the kitchen display. On the other hand, if the kitchen display mirrors the other TV content, then the two displays would change in tandem.

For combining sources, this could be “picture-in-picture” (which some TVs offer today as a feature), different sources for the audio and video, overlaid video (less common), or tiled content (as security camera systems usually offer).

Figure 3 shows a basic stream (solid) with a number of possible ways to hang additional sources or additional sinks. Note that if the extreme left dotted source is included, then the next source is also an intermediate. Similarly, if the extreme right dotted sink is included, the sink to its left is also an intermediate. When a stream wakes, there is the question of whether the stream should wait to become active until all sources and sinks are ready, or when at least one source and at least one sink is ready (this can be determined on a local level — it does not need to be standardized).

As long as the structure of the stream remains unchanged, no new complications are introduced by multiple sinks or sources, but if the combination of involved devices changes, then some devices may need or want to power up or down as a consequence. This is the subject of the next section.

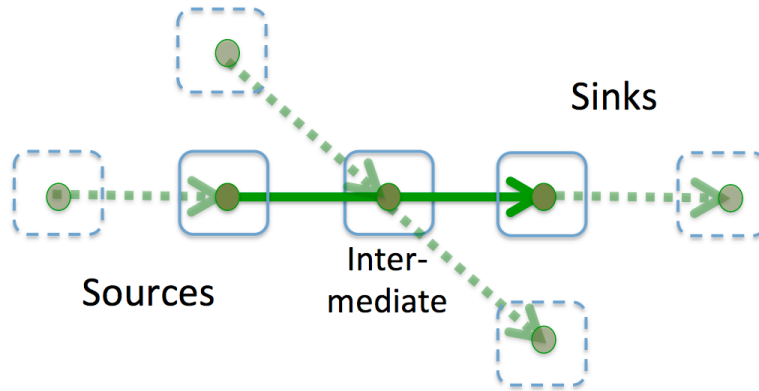


Figure 3. Stream topologies with multiple sources or sinks

Creating streams

Streams are only created when all the involved devices are awake, so that this activity does not involve sleeping devices. This simplifies device operation while asleep. There does need to be a stream created for each input or output on a device, so that a wake signal can be relayed to the connected device when a specific input or output is selected.

Changing stream structure

There are two ways to implement waking streams:

- the stream has a single existence which is transitioned from a sleep state to an on state;
- a copy of the sleeping stream is created but has a fate distinct from the original sleeping one.

It seems likely that both of these are useful so both should be supported by the technology. When a stream is created, it will be tagged as “static” or “dynamic” (better terms needed but these are fine for now).

Consider a stream A-B, where A is a source and B a sink. For a static stream, when it wakes, a copy is created as the active stream, and if a second sink C is added as a mirror of B, it becomes A-B-C. The original remains A-B and when the stream goes to sleep, the changed version is discarded. For a dynamic stream, adding C makes the underlying stream A-B-C so that when it goes to sleep, C remains as part of the stream.

Failure

A behavior may encounter a problem that causes the normal outcome to not occur. In these cases, it needs to be communicated to the user what is going on, and a set of fallback actions needs to occur. One clear way that failure can occur on starting a stream is why one of the needed devices is off (not asleep), and so can't wake to play its role. Another failure scenario occurs when a needed device rejects a waking stream; a common circumstance for this would be if it is already busy with a stream and cannot or does not want to do both (or stop/pause the first stream).

When a stream initiation fails, then any device involved that can signal an error mode should do so, explaining as much as it can. This condition should remain for a period of time (perhaps 10 minutes) to enable the user to understand what is going on and potentially take some action to remedy the situation. If the time passes with no further action, then devices showing the error can go back to sleep.

Consider a situation in which a stream is woken, but the sink (display) part of the stream is off. This will cause an error which may then lead the user to turn the sink on to enable the stream to operate. When the display comes on, it should not first look to the last stream it was part of, but rather see if any active streams are queued up.

In the general case, if a wake is initiated by addressing a device, the device may default to restarting the last stream in which it participated. However, if the device is woken through the stream, then the waking stream should take priority.

Occupancy sensors

In recent years, a few TV and computer monitor models have been introduced that include occupancy sensors. The most common likely usage of these is to trigger a sleep mode when the device believes that no viewer is present for a period of time. Since the sensor may not always be correct, a usual mode of operation would be to display an overlay image that the device will power down soon if no activity is detected. This might be left up for some small number of minutes. Some devices do this today. This is use case 1-4.

Some people routinely just listen to a TV, e.g. from a different room, for extended periods of time. Thus, powering down the audio may be governed by a different algorithm than that used to power down the video. This also raises the question about how to overlay a warning on audio that the stream is about to power down.

Over time, we should expect to see occupancy sensors on devices and in rooms that are not necessarily on the display itself. There may be no sensor on the display, or external sensors (on other devices) may provide more reliable data.

Emergency broadcasts

Some countries have long had facilities for emergency broadcast signals, both introduced into ongoing broadcasts (TV and radio) as well as special receivers that can power up on receipt of a signal that an emergency broadcast is imminent. With digital devices, emergency broadcasting becomes even easier and more effective, and can better target specific locales. Determining who should have the authority to send such broadcasts, and how to securely send them, are difficult issues but outside the power control scope.

It would be easy to simply create one or more streams for emergency broadcasts that would wake devices as needed when they occur (devices that are off will generally not be on the network so cannot be turned on remotely). The source device in the local network would have a way of being alerted by the emergency broadcasting entity. People could then choose which displays receive the broadcast in a building. Emergency streams could be tagged as such and

might have special semantics, e.g. always dropped in as picture-in-picture when a display is already busy with a different stream, or causing a pause in a fixed stream, etc.

Diverging audio and video

A conventional AVR splits the audio and video signals to display the audio at attached speakers and send the video to the TV (technically, the audio may be still transmitted to the TV, but when separate loudspeakers are present the in-TV speakers are usually not used). For power control, this is equivalent to any split stream; that one part of it is lacking a video component and another part is lacking an audio component does not change the semantics of stream operation. An entire stream can be audio-only or video-only.

All that said, there is obvious advantage to not transmitting data that is not being used. If a sink is capable of only audio or of only video, or is only displaying one medium at a particular time, then it can be helpful to only transmit that one medium to the sink (this would not necessarily apply to an intermediate which also has a sink function, as downstream sinks might want the other medium).

When a typical A/V stream has its audio muted, it will usually make sense to continue to deliver the audio, at least for a short time, so that when it is unmuted the audio can resume as quickly as possible.

Sleeping intermediate devices

Some intermediate devices sold today can pass an HDMI signal along while in a sleep mode. In this scenario, the intermediate device is not required to wake up; it is essentially acting like an IP network infrastructure device by passing along data without interacting with it.

HDMI switches

The HDMI protocol supports, and one can buy, an “HDMI Switch.” The simplest ones operate like an AVR in being able to select one of several inputs and route it to the output. More complicated ones can duplicate the output on multiple ports, or route several independent streams from arbitrary inputs to arbitrary outputs. These switches do not seem to introduce any new functionality or needed behaviors. Each stream is logically separate.

Legacy Devices

A final consideration that could introduce new behaviors is “legacy” devices that do not implement the sleeping stream concept above, and may not have the stream concept explicit in their operation at all (it is inherently implicit in any A/V device). This would include all existing A/V devices, excepting those that could be upgraded to include the new functionality (these called “compliant” in this document) via a software update. Some legacy devices have digital communications ability, but simply less than fully compliant devices. Other legacy devices continue to use analog connections and can not communicate at all (except perhaps via remote control commands and similar technologies). Because of this, information in the

network is imperfect, some desired signaling or control capabilities are absent, and power control is necessarily “best effort.”

Legacy devices can be still represented in streams, and marked accordingly. Some may be able to communicate with individual compliant devices, directly or indirectly, and the stream could be tagged with knowledge of which compliant devices have particular communication abilities with which legacy ones. Most commonly it will be the compliant device connected directly to the legacy device that has some communication ability with it. A legacy HDMI device can relay information about its power and functional state to a compliant one, and commands about power and functional state can be sent from the compliant device to the legacy one. The compliant device can essentially compute what the legacy device would do if it implemented sleeping streams; the compliant device then commands the legacy device accordingly. Even analog connections offer some capability; for example, a compliant AVR can sense a signal coming into it from a device that does not otherwise communicate and infer that it is On and sending a signal (this could then cause a stream to wake). Signals sent to non-compliant devices over analog links offer little or no opportunity for communication.

Legacy devices are easiest to understand when at an end of a stream, with compliant devices in the middle. It is possible to have a legacy device in the middle, but we do not consider that here. The compliant device to which a legacy device is connected can monitor data and notice when the legacy device is present or absent, and wake or put to sleep the relevant stream, as events from the source suggest. For a legacy sink, if the compliant device notices that the sink changes power state, it can then act on that information (though often the changing power state of a legacy sink will not be apparent to the compliant device).

For some purposes, the non-compliant devices can be seen as external to the local network, in the way that IP devices elsewhere on the Internet are treated. The compliant device that is connected to the non-compliant one is then the effective local source or sink. This then folds into the general model for devices outside the local network, with the stream only accounting for local ones. An exception to this is when a device in the local network OTHER THAN the one directly connected to the non-compliant one has communication with or control of the legacy device. For example, there are “IR blaster” and 12 V control technologies in some systems for devices to command arbitrary other devices, across data paths other than the stream path. More generally, if a compliant AVR has several legacy devices connected to its inputs, then at least one stream needs to exist for each input so that when the AVR is woken it knows which input is the one to use. Thus there are multiple reasons to model the legacy devices in the stream.

Summary

This document has reviewed concepts involved in A/V streams, use cases for managing devices and streams, resulting behavior models, and other considerations. While the number of use cases considered is large, the actual number of behaviors that any individual device needs to implement is much more modest. That is, the complexity introduced with sleeping streams appears to be quite manageable, and additionally, this avoids devices having to command each

other, removing some complexity. A next step is to consider what would be needed to add these capabilities to existing technologies such as HDMI and IP-based protocols, and further consideration of user interface issues.

Acknowledgement

Many thanks to Gari Kloss for her invaluable contributions.

References

- Nordman et al. 2007, Draft Principles for Energy Efficient Digital Networks and Network-connected Devices, International Workshop on Energy Efficient Set-top Boxes and Digital Networks, International Energy Agency, July 6, 2007. Paper by Bruce Nordman, Alan Meier, Mark Ellis.
http://www.iea.org/Textbase/work/workshopdetail.asp?WS_ID=285
- Nordman, Bruce, 2012A. Existing Product Assessment and Conclusions Memo, interim deliverable for this project, June 15, 2012.
- Nordman, Bruce, 2012B. Summary of Analysis of Communication Link Technologies, interim deliverable for this project, September 21, 2012